Tip of Your Tongue: Methods for an Effective Reverse Dictionary Model

Stanford CS224N Custom Project. TA mentor: Ben Newman

Arman Aydin
Department of Computer Science
Stanford University
aaydin06@stanford.edu

Cem Gokmen
Department of Computer Science
Stanford University
cgokmen@stanford.edu

Abstract

A reverse dictionary is a tool that finds a desired term given its rough description or definition. We propose the use of sequence-to-sequence Transformer models to perform a reverse dictionary lookup, as well as task-specific data augmentation and diversity-based generation methods to improve performance. We show that our suite of data augmentation and diverse generation methods together allow us to reach up to 34% top-10 and 47% top-20 accuracy on our hand-crafted test set, compared to 10% using a baseline trained with non-augmented training data and a naive beam search generation method.

1 Introduction

A reverse dictionary is a mechanism that can find a desired target term given a rough description or definition [1]. Table 1 shows an example reverse dictionary query.

Input	Prediction
"a tall, long-necked, spotted ruminant of Africa"	**"giraffe"**

Table 1: An example for the reverse dictionary task.

A reverse dictionary tool is widely used among writers, translators, and language-learners for finding a suitable term for their desired meaning. It is especially useful in **tip-of-the-tongue** situations, where one can remember a description or the synonyms of the term, but not quite recall the exact words.

The main challenge behind reverse dictionary is to match the description given by the user to a word semantically. Indeed, recent neural models can encode the description and the words into the same semantic embedding space, returning the closest word to the given description [2]. Previous model architectures like Recurrent Neural Networks (RNN) and Long-Short Term Memory (LSTM) networks had difficulties during this encoding due to a lack of high-resource dictionary and word datasets and methods involving the use of static word embeddings [1].

With the use of Transformer[3]-based models, these issues have been averted. Since Transformer models are highly rich in representation due to being trained over a large corpus during pre-training, the Transformer models can relate the descriptions to target words better. Additionally, Transformer models like BERT can output contextualized representation for a word, which deals with the problem of static embeddings [2]. However, even though the Transformer-based approaches give a large advantage due to pre-training over previous models, naively fine-tuning on raw dictionary data will cause the models to run into various problems in real-world scenarios.

To this end, we propose and analyze a reverse dictionary setup using an off-the-shelf sequence-to-sequence Transformer, a task-specific data augmentation method that generates queries using word hierarchy, and a diverse sampling mechanism to be able to generate diverse predictions efficiently. We show that our setup exhibits strongly better performance compared to a naively-trained and sampled Transformer baseline.

2 Related Work

Previously, the **reverse dictionary** task has been investigated with various neural network architectures. RNNs were trained with dictionary definitions or sentences from the encyclopedia, where the goal of the model was to map these definitions to the target words [1]. Due to the dependency shortcomings of RNNs, LSTM models training on larger datasets like Wiktionary [4] and WordNet [5] were proposed to achieve higher performance [1]. The performance of LSTM models were significantly better than RNNs, where the performance increase is accredited to the quality of the Wiktionary and WordNet datasets. As a commercial reverse dictionary resource, *OneLook.com* is an algorithm that searches 1061 indexed dictionaries.

Recently, as Transformer-based models proved to be exceptionally powerful in many NLP tasks, the focus on models shifted towards fine-tuning various pre-trained transformer models like *BERT* and *T5* [6]. However, most works used BERT as an encoder in their downstream NLP tasks like dependency parsing [7] and named entity recognition [8], and less frequently for generation tasks. Recently, work on generation through sampling sentences from BERT have proved that BERT's potency in generation tasks [9]. Similarly, initialized as an unsupervised machine training model, pre-trained BERT achieved good performance in recent studies [10].

The promising generative performance of Transformer language models created a foundation for reverse dictionary tasks, where BERT was further investigated through creating monolingual and cross-lingual reverse dictionaries that performed better than previous reverse-dictionary models [2]. Moving forward, a reverse-dictionary application that is true to its real-world use cases under what a user would input rather than a synthetic dictionary definition is important to explore, especially with the performance improvements transformers provide.

3 Approach

3.1 Problem Definition

We define the reverse-dictionary problem as a sequence-to-sequence supervised learning problem: given a query (a sequence of words), we expect it to output the predicted term (a short sequence of words). We propose using a Transformer-based sequence-to-sequence encoder-decoder architecture, whose examples we discussed above, and which are known to perform well even on more complicated tasks such as text summarization. We propose training our model directly in a supervised learning manner where term-definition pairs from a dictionary are reversed to become definition-term pairs, and fed into the model as the inputs and labels.

3.2 Base Model

For our base model, in light of the promising performance of Transformers, we look at different Transformer models to fine-tune on the reverse dictionary task. BART [11] is one such recent transformer model, with a bidirectional encoder and an auto-regressive decoder, that has been widely used for various generation tasks. It is pre-trained over corrupted text with an arbitrary noising function by learning a model to reconstruct the original text.

Similarly, T5 [12] has been presented as a strong Transformer model that is particularly effective as a *few-shot learning* system, where prepending a task description corresponding to a task like *"translate from English to French"* or *"summarize:"* allows the model to perform different tasks. This model is also a sequence-to-sequence encoder-decoder model that was pre-trained over a multi-task mixture of unsupervised and supervised tasks. Most notably, teacher forcing, which is done by inputting a sequence and correcting the model subject to a corresponding target sequence, was used during its training.

3.3 Data Augmentation

Beyond simply training the model purely on dictionary definition-term pairs, which are typically very formal definitions that do not necessarily match the kinds of queries a user might pose to a reverse dictionary, we also apply data augmentation using additional hierarchy data found in dictionary datasets:

- For objects that have synonyms defined, we generate additional queries in the below forms:
  1. *a word similar to {comma-separated synonyms}*
  2. *a synonym of {comma-separated synonyms}*
- For objects that have antonyms defined, we generate additional queries in the below forms:
  1. *the opposite of {comma-separated synonyms}*
  2. *an antonym of {comma-separated synonyms}*
- For objects that have their parts of speech labeled, we generate additional queries in the below forms:
  1. *a {part of speech} meaning {definition}*
  2. *{definition}, {part of speech}*

We postulate that these data augmentation methods will have very positive impact on model performance by allowing the model to train using data that is closer to the in-the-wild user query distribution as opposed to artificial queries produced simply by reversing the dictionary entries.

3.4 Sampling Methods

In most cases, the term the user is searching for will not be the result of the highest beam obtained from beam search on the model. If the beam search is widened to output the results of the top K beams where $k > 1$, a typically observed phenomena is that the model will output similar forms of the same word. For example, for the query *the discipline of creating buildings* (which is intentionally vague, as an example), the words *construction, constructional, construct, constructor* can be found by beam search before *architecture*, but it is more useful to output the latter in this case, since finding different forms of the same word is typically not valuable in the reverse dictionary context (e.g. the root word is typically what the user is looking for rather than a particular form). As a result, one of the most important issues with a reverse dictionary model is being able to generate sufficiently *diverse* predictions using the model.

To this end, we postulate that the Diverse Beam Search generation method from [13], which splits beams into groups during the beam search phase, and encourages diversity by adding to the beam search objective a dissimilarity term (e.g. a similarity penalty) that penalizes a beam for being similar (e.g. containing the same tokens as) beams from other groups.

4 Experiments

4.1 Training Data

For training data, we use a dump of all of the words and definitions found in Wiktionary [4]. The dataset contains 1.3 million word senses. To sanitize the data, we first remove words marked with the part of speech "*name*". These words are often proper nouns that are poorly defined (e.g. "California" as "a place in the United States"). Additionally, for words that appear to have multi-line definitions, we use only the first line of the definition, which heuristically appears to be the definition rather than example sentences etc. that were mistakenly placed in the definition field. We randomly split this dataset into training and validation sets at 95% and 5% of the full dataset respectively. In the end, 982,736 definitions for 875,490 unique words were used for training and 51,779 definitions for 46,077 unique words for validation. With the augmentation methods discussed above, the training set grew to 2,948,207 data points for training.

Additionally, we use a dump of all words from the Open English WordNet [14], which is an open-source expansion of the classical WordNet [5] dataset. The WordNet dataset builds graphical representations through lexical and semantic similarities across words, categorizing words with

similar linguistic structures in clusters, as well as providing word definitions. 201,182 definitions were obtained from this dataset. With the augmentation methods discussed above, this number was increased to 549,087 data points for training. Performance statistics are separately reported for experiments that used this dataset along the Wiktionary one.

4.2 Test Data

Besides the validation set, which was sampled from the Wiktionary data and thus follows the distribution of dictionary definitions rather than reverse dictionary queries, we present an original test dataset that was created for model evaluation using a random sample of 100 words from the English dictionary. The sampled words were divided into 4 scenarios that simulate possible real-life query formats a user might give to a reverse dictionary. For the first group (30 words), some description of the word and its part of speech was provided as a query by a human expert. For the second group (30 words), only the description of the word was provided as a query by a human expert. For the third group (20 words), the word's synonym or antonym was provided as a query by a human expert. Finally for the fourth group (20 words), a set of 5 words that are related to the word were provided as a query by a human expert. Notice that none of the queries were formed through the dictionary but through possible real-life human queries to replicate the queries it might face in the case of deployment, thus presenting a better opportunity at evaluating model performance on a more representative sample of the real distribution of possible reverse dictionary queries.

Some examples from the test set are reported in Table 3 in the Analysis section.

4.3 Evaluation method

For our baseline, we use *top-k accuracy* as our evaluation method. For the validation and test datasets, we produce k predictions for each input, and we mark the prediction as "correct" if at least one of the predictions exactly matches the expected word. We compute accuracy by dividing the number of correct predictions by the number of total queries. For the charts and accuracy numbers discussed in this paper, top-10 accuracy evaluation was used unless otherwise stated.

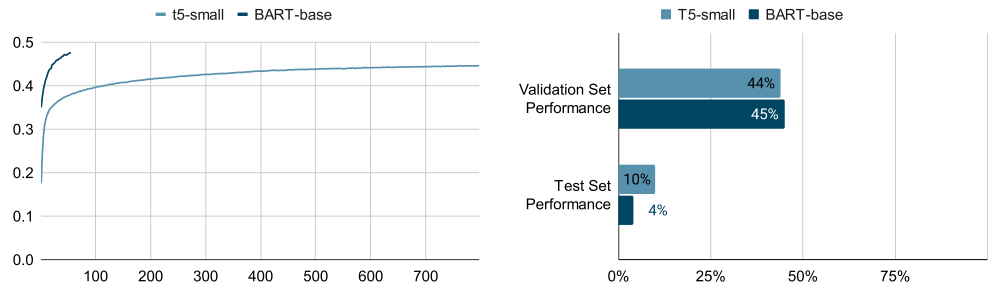
4.4 Experimental details

To run these experiments, the HuggingFace Transformers [15] library was used on a setup consisting of 6 Nvidia RTX 2080 Ti GPUs. The default training configurations were used for all models. A learning rate of 2×10^{-5} and a learning rate decay of 0.001 was used. We also report the training times for the baseline models: the models were both trained for 4 hours due to budgetary constraints - this meant training for fewer steps on the BART model due to its increased parameter count. Overall, the BART model was trained for 4 epochs and the T5 model for 10 epochs. When diverse beam search is used for generation, we use a diversity penalty of 10.0 and 5 groups of beams. We also ban the presence of identical 3-grams in different beams.

4.5 Results

4.5.1 Baseline

To choose our baseline, we experimented with fine-tuning pre-trained BART-base and T5-small models on the non-augmented Wiktionary dataset. The two models have drastically different parameter counts 140 million and 60 million respectively: they were chosen this way so that the same experiment could also act as a proxy for understanding the trade-offs between gaining expressiveness from more parameters vs. being able to train for longer with a smaller model. In Figure 1 we report the results of these experiments.



(a) Validation accuracy (y-axis) after training with each baseline candidate by training batches (x-axis)

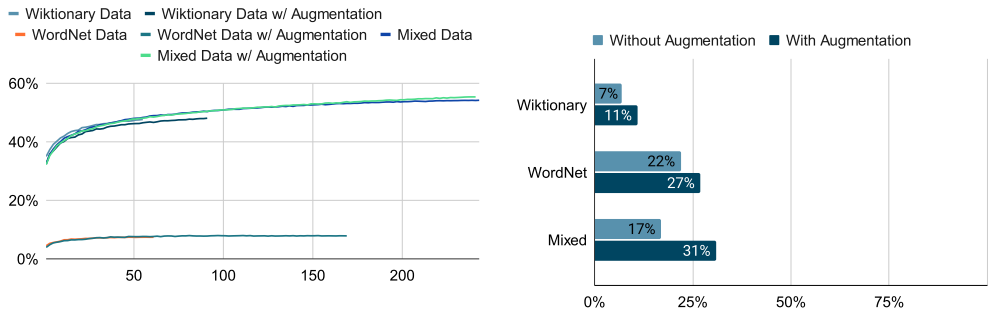
(b) Validation and test set accuracy after training with each baseline candidate

Figure 1: Loss and accuracy curves from the baseline training process

The results indicate **BART-base** to be the superior baseline at an equal wall-clock training time (despite training for significantly fewer epochs), and thus it is used as the baseline for all of the further experiments. They also establish the difficulty of the problem: the distribution of the test set and the distribution of the default Wiktionary training set must not be very similar, leading to a clear discrepancy between validation and test set performances, and a truly abysmal test performance at 10% that we can improve upon.

4.5.2 Training Data and Augmentation Method

Then, building upon the baseline, we retrain using our different training dataset combinations: one of {Wiktionary, WordNet, Mixed} for the data source, and one of {Augmented / Non-Augmented} for choice of data augmentation methods. We report the results of this experiment in Figure 2.



(a) Validation accuracy (y-axis) after training with each dataset by training batches (x-axis)

(b) Test set accuracy after training with each dataset

Figure 2: Loss and accuracy curves from the multi-dataset training process

Here we note that adding data from the WordNet dataset increases the test accuracy to 17% compared to the baseline 10%. Applying additional data augmentation in the form of synonym, antonym, and part-of-speech queries further increases accuracy to 31%, already a clear improvement that starts to show acceptable performance.

4.5.3 Generation Method

Then, building upon the mixed-and-augmented dataset results, we investigate the use of diverse beam search for generation, and its effect on the model's test set accuracy. In Figure 3 we report the results of this experiment.

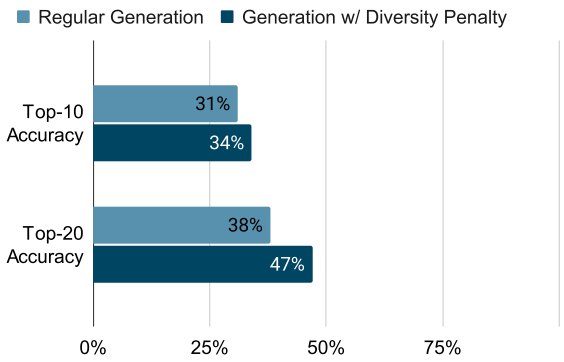


Figure 3: Test set accuracy using different generation methods and different K for top-K metrics

The results strongly suggest that keeping diversity in mind during generation is critical to model success. Doing so increases top-10 accuracy from 31% to 34%, and the effect of diverse generation is even more pronounced when reporting top-20 accuracy, rising from 38% to 47% (e.g. being able to answer nearly half of the queries correctly).

Overall, our experiments clearly show the benefits of data augmentation and diverse beam search to the reverse dictionary task.

5 Analysis

Beyond reporting just performance numbers, we select a sample of test set entries, run these through or model and analyze the results. We identify the following error types:

Error Type	**Description**
S	The model successfully predicted the label within its top 10 generated sequences.
EE	Wrong due to evaluation error: the model successfully predicted a form of the label within its top 10 generated sequences, but the form did not match the lemma and was marked as incorrect.
A	Wrong due to model error: the model failed to predict a clear response to the query in the absence of any confounding factors.
B	Wrong due to limited number of predictions in a context with many similar words.
C	Wrong due to generation errors, i.e. repeating different forms of same root, etc.
D	Wrong due to training distribution imbalance: a hyponym of the target word was predicted.

Table 2: A description of different types of errors seen in our test data.

Then, in Table 3, for each element in our test set sample, we report the user query, the model's predictions, the expected answer, and the error type we believe that query maps to. We also present some success examples.

User Query	Model Predictions	Expected Answer	Err. Type
a noun meaning a thick and usually shiny fabric	silk, sparkling, glossy, glaze, satin, palladium, pall, sparkle, cloak, jewelry, lacquer, satinny, gaudier, coffin, chiffon, satinette, taffeta, tinsel, briquet, bronze	velvet	**B**
a noun meaning a governing body consisting of multiple people	mixed council, congregation, congress, "peoples council", commons, puppet government, body of government, superarchy, council, corporation, multi-party, collective council, body of representatives, "peoples assembly", joint council, plural government, corps, panarchy, chamber, kingdom	council	**S**
a verb meaning to break, rip, in a loud way, especially for rope, cables, etc	ramp, rope, crash, ripe, tear, snarl, rattle, louse, razzle, splash, hurl, crack, jerk, hurtle, loud, whip, bust, hail, jaw, bang	snap	**A**
a verb meaning to underline or put focus or importance on something	emphasize, emphasise, focalize, focus, focalise, concentrate, underpin, lighten, magnify, lighten up, flatten out, highlight, pin down, narrow down, bolster, embolden, sharpen, overfocus, light up, inflect	emphasize	**S**
people who are richer or more prestigious than others	big money, rich, prestigious, superrich, predominate, rich people, earners, upper class, prestige, upper crust, grandfathers, earnership, high power, preeminent, superiority, behemoths, world class, higher power, lower class, gilded society	elite	**A**
something that cannot be seen	unseeable, blindness, obscure, blind spot, shadow, darkness, invisible, vision, insight, hideaway, unseeability, eyeball, discernment, invisibility, obscurity, eyebrow, visibility, incognita, nakedness, unsee	invisible	**S**
similar to specific	similar, specific, descriptive, specific, similarity, subspecific, identical, aspecific, similarish, similaristic, similarlike, distinctive, specifical, specificistic, sociose, specificlike, synonymous, sensory, approximate, sensational	particular	**C**
similar to regular	regularlike, regularish, regular-like, irregular, regularoid, regularistic, rhyme, regulariform, regularous, uniform, regulary, regular, regular-looking, semiregular, regularical, nonregular, regular-sounding, isomeric, regularist, regularwise	general	**C**
opposite of red	blue, white, green, yellow, black, red, purple, bloody, brown, orange, reddish, blood, grey, olive, naked, blue, bright, coloured, ruddy, blonde	green	**S**
school, university, teenager, freshman, kid	junior, senior, schooly, kiddie, collegiate, sophomore, yo-yo, youth, kiddo, college kid, schoolling, schooer, highschooler, youthling, yuppie, youthie, pup, newcomer, schoole, schooner	student	**D**
anxious, nervous, sweat, unexpected, sad, future	worried, doubtful, upset, fearful, expectant, troubled, nervous, afraid, frustrated, disconcerted, sadistic, fearsome, excited, disheartened, puzzled, uncertain, unsettled, anxious, nosey, fretful	worry	**EE**
risk, return, gamble, stock, company	risk-off, wager, bet, gamble, risk-on, stake, brokerage, dope, risk-ass, risk-up, go-go, warrant, barnacle, punt, venture, gambol, bargain, poker, risky, adventure	stake	**S**

Table 3: A sample of test-set queries, model predictions, and expected answers, categorized by the type of error made.

6 Conclusion

To conclude, our experiments show that:

1. Without the use of data augmentation and diverse generation methods, model top-10 accuracy remains extremely low at 7%.
2. Adding data from the WordNet dataset increases the accuracy to a still-low 17%.
3. Applying additional data augmentation in the form of synonym, antonym, and part-of-speech queries increases accuracy to 31%.
4. Applying generation with diversity penalty increases top-10 accuracy to a further 34%, and shows an even more drastic improvement in top-20 accuracy to 47%.

As a result, we show that combining a modern sequence-to-sequence Transformer model with purpose-built data augmentation and diverse sampling methods drastically in allows us to train a reverse dictionary model that can be used effectively on in-the-wild user queries.

References

[1] Felix Hill, Kyunghyun Cho, Anna Korhonen, and Yoshua Bengio. Learning to understand phrases by embedding the dictionary. *Transactions of the Association for Computational Linguistics*, 4:17–30, 2016.

[2] Hang Yan, Xiaonan Li, and Xipeng Qiu. BERT for monolingual and cross-lingual reverse dictionary. *CoRR*, abs/2009.14790, 2020.

[3] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2017.

[4] Wiktionary. Wiktionary, the free dictionary, 2021. [Online; accessed 9-February-2022].

[5] George A. Miller. WordNet: A lexical database for English. In *Human Language Technology: Proceedings of a Workshop held at Plainsboro, New Jersey, March 8-11, 1994*, 1994.

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018.

[7] Hang Yan, Xipeng Qiu, and Xuanjing Huang. A unified model for joint chinese word segmentation and dependency parsing. *CoRR*, abs/1904.04697, 2019.

[8] Xiaonan Li, Hang Yan, Xipeng Qiu, and Xuanjing Huang. FLAT: chinese NER using flat-lattice transformer. *CoRR*, abs/2004.11795, 2020.

[9] Alex Wang and Kyunghyun Cho. BERT has a mouth, and it must speak: BERT as a markov random field language model. *CoRR*, abs/1902.04094, 2019.

[10] Guillaume Lample and Alexis Conneau. Cross-lingual language model pretraining. *CoRR*, abs/1901.07291, 2019.

[11] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *CoRR*, abs/1910.13461, 2019.

[12] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *CoRR*, abs/1910.10683, 2019.

[13] Ashwin K Vijayakumar, Michael Cogswell, Ramprasath R. Selvaraju, Qing Sun, Stefan Lee, David Crandall, and Dhruv Batra. Diverse beam search: Decoding diverse solutions from neural sequence models, 2016.

[14] John Philip McCrae, Alexandre Rademaker, Ewa Rudnicka, and Francis Bond. English WordNet 2020: Improving and extending a WordNet for English using an open-source methodology. In *Proceedings of the LREC 2020 Workshop on Multimodal Wordnets (MMW2020)*, pages 14–19, Marseille, France, May 2020. The European Language Resources Association (ELRA).

[15] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Huggingface's transformers: State-of-the-art natural language processing, 2020.